

REMARKS

Claim amendments

Claims 1, 7 and 13 have been amended to more particularly point out and distinctly claim the invention and distinguish over the MATLAB user's guide, cited in the office action.

In particular, the claims have been amended to more particularly recite that the mask (the second image) is applied to the first image via a logical operation which results in deletion of areas peripheral to said region of interest and leaves pixel values for said region of interest undisturbed. For support for this amendment, and the newly added claims, see the specification at pages 6, 12 and 14, and Figures 3-6, particularly the references to the description of the logical AND operation as the means for applying the mask to the image.

Anticipation rejection

In the office action, all claims were said to be anticipated by the Matlab User's Guide. Applicants request withdrawal of the rejection.

Claims 1, 7 and 13 are not anticipated by the MATLAB User's Guide because the Guide does not disclose the application of the mask to the image as recited in step f) of claims 1 and 7 or paragraph 6) in claim 13.

These claims require a step or function of applying the mask to the first image in a logical operation wherein application of said mask to said first image via the logical operation deletes areas peripheral to said region of interest and leaves pixel values for said region of interest undisturbed. No such teaching is disclosed in the Matlab Guide.

In the portion cited by the Examiner (pages 11-7 and 11-8 in conjunction with page 14-346) the reference is describing a filtering operation in which a filter is applied to a region of interest in an image, with the mask defining the region of interest. However, in the text at pages 11-7 and 11-8) the areas outside of the region of interest are not changed at all, and the filtering operation is performed in the region of interest, that is, the area within the mask the pixel values are changed.¹ This is *precisely the opposite of what is recited in claims 1, 7, and 13-- the areas peripheral to the region of interest are deleted, and the areas in the region of interest are undisturbed.*

Page 14-346 of the User's Guide describes the `roifilt2` filtering operation of pp. 11-7 and 11-8, namely filtering values for pixels where the mask contains 1's (white area of the mask) and returning unfiltered values for pixels in locations where the mask contains 0's (the black areas). This is what is shown in the example with the girl's jackets on pages 11-7 and 11-8 - filtering, i.e., disturbing pixel values in the region of interest and leaving portions out the region of interest unchanged, the antithesis of what is claimed in the claims.

In view of the above, it is clear that the Matlab User's Guide does not anticipate the claims. The rejection should be withdrawn.

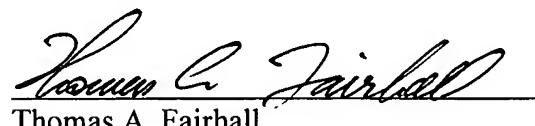
¹ The applicant has found a more recent version of the User's Guide on the Web which basically is the same as pages 11-7 and 11-8, but with the benefit that the illustrations on these pages can be seen clearly. These pages are enclosed for the convenience of the Examiner. It can be seen that the bird logo on the girl's jacket in the picture (the region of interest specified by the mask per page 11-7) is filtered in a manner to make it sharper or more clearly seen, whereas the areas outside of the mask (face, background, etc.) are unchanged.

Favorable reconsideration of the application is requested.

Respectfully Submitted,

McDonnell Boehnen Hulbert & Berghoff LLP

Date: January 3, 2007 By:


Thomas A. Fairhall
Reg. No. 34,591
Telephone 360 379 6514

CERTIFICATE OF MAILING

The undersigned hereby certifies that the foregoing RESPONSE is being deposited as first class mail, postage prepaid, in an envelope addressed to Mail stop Amendment, Commissioner for Patents, P.O. Box 1450 Alexandria VA 22313-1450 on this 3rd day of January, 2007.


Thomas A. Fairhall

Filtering a Region

Filtering a region is the process of applying a filter to a region of interest in an image, where a binary mask defines the region. For example, you can apply an intensity adjustment filter to certain regions of an image.

To filter a region in an image, use the `roifilt2` function. When you call `roifilt2`, you specify an grayscale image, a binary mask, and a filter. `roifilt2` filters the input image and returns an image that consists of filtered values for pixels where the binary mask contains 1's and unfiltered values for pixels where the binary mask contains 0's. This type of operation is called *masked filtering*.

Note `roifilt2` is best suited to operations that return data in the same range as in the original image, because the output image takes some of its data directly from the input image. Certain filtering operations can result in values outside the normal image data range (i.e., [0,1] for images of class `double`, [0,255] for images of class `uint8`, and [0,65535] for images of class `uint16`). For more information, see the reference page for `roifilt2`.

Example: Filtering a Region in an Image

This example uses masked filtering to increase the contrast of a specific region of an image:

1. Read in the image.

```
I = imread('pout.tif');
```

2. Create the mask.

This example uses the mask `BW` created in [Selecting a Polygon](#). The region of interest specified by the mask is the logo on the girl's jacket.

3. Create the filter.

```
h = fspecial('unsharp');
```

4. Call `roifilt2`, specifying the image to be filtered, the mask, and the filter.

```
I2 = roifilt2(h,I,BW);
imshow(I)
figure, imshow(I2)
```

BEST AVAILABLE COPY

Image Before and After Using an Unsharp Filter on the Region of Interest

Specifying the Filtering Operation

`roifilt2` also enables you to specify your own function to operate on the region of interest. This example uses the `imadjust` function to lighten parts of an image:

1. Read in the image.

```
I = imread('cameraman.tif');
```

2. Create the mask. In this example, the mask is a binary image containing text. The mask image must be cropped to be the same size as the image to be filtered.

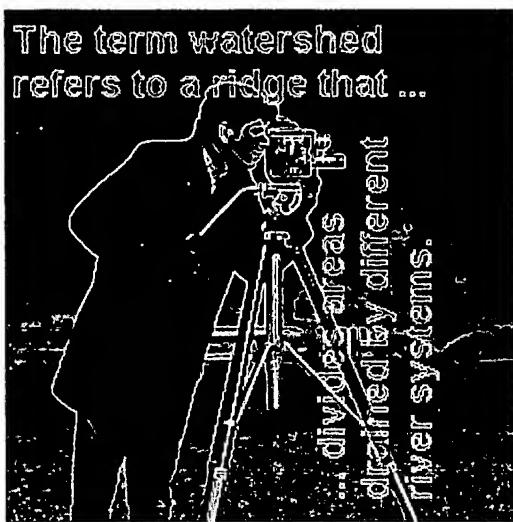
```
BW = imread('text.png');
mask = BW(1:256,1:256);
```

3. Create the filter.

```
f = inline('imadjust(x,[],[],0.3)');
```

4. Call `roifilt2`, specifying the image to be filtered, the mask, and the filter. The resulting image, `I2`, has the text imprinted on it.

```
I2 = roifilt2(I,mask,f);
imshow(I2)
```

BEST AVAILABLE COPY**Image Brightened Using a Binary Mask Containing Text**

Other Selection Methods

Filling a Region

© 1994-2005 The MathWorks, Inc.